OCR BASED MOBILE APPLICATION

A Mini Project Report

submitted by

KT VIMAL(MES20MCA-2024)

to

the APJ Abdul Kalam Technological University
in partial fulfillment of the requirements for the award of the Degree

of

Master of Computer Applications

Department of Computer Applications

MES College of Engineering
Kuttippuram, Malappuram - 679 582

February 2022

DECLARATION

I undersigned hereby declare that the project report **OCR Based Mobile Application**, submitted for partial fulfillment of the requirements for the award of degree of Master of Computer Applications of the APJ Abdul Kalam Technological University, Kerala, is a bona fide work done by me under supervision of Mr. Syed Feroze Ahamed.M, Assistant Professor, Department of Computer Applications. This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Place:KUTTIPPURAM

KT VIMAL(MES20MCA-2024)

Date:04-03-2022

DEPARTMENT OF COMPUTER APPLICATIONS
MES COLLEGE OF ENGINEERING, KUTTIPPURAM

CERTIFICATE

This is to certify that the report entitled **OCR Based Mobile Application** is a bona fide record of the Mini Project work carried out by **KT VIMAL(MES20MCA-2024)** submitted to the APJ Abdul Kalam Technological University, in partial fulfillment of the requirements for the award of the Master of Computer Applications, under my guidance and supervision. This report in any form has not been submitted to any other University or Institution for any purpose.

Internal Supervisor(s) External Supervisor(s)

Head Of The Department

Acknowledgements

First of all I thank and praise god almighty, for giving me the strength and power to complete my project work without any hindrance and for molding me into what I'm.

With great respect I express my sincere thanks to Dr.K.A Navas, Principal, MES College of Engineering, providing facilities for this project. I extend our heartfelt thanks and gratitude to Mr. Hyderali.K, Associate Professor, Head of the Department of Computer Applications.I would also like to express my gratitude towards my supportive and encouraging project coordinator, Ms. Priya J.D, Assistant Professor, Department of Computer Applications.I have great pleasure to express my gratitude to Mr. Syed Feroze Ahamed.M , Assistant Professor, internal guide for his valuable suggestions and encouragement. I extend our sincere thanks to all my teachers in the department for their constant encouragement and never ending support throughout the project. Lastly by no means the least, I shall be amiss in not proffering my loving thanks to my dear parents and friends for their encouragement, support, cooperation and good wishes, without which I would have been very sorely tested in leading this project to its successful completion. Well-wishers also gratefully acknowledged.

KT VIMAL(MES20MCA-2024)

Abstract

OCR stands for Optical Character Recognition. It is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents and images. OCR software is able to go through documents and make the contents machine-readable, so that they can be worked with in an electronic format. It can be used for many different documents and allows many tasks to be automated.Significant number of algorithms is required to develop an OCR and basically it works in two phases such as character and word detection. In case of a more sophisticated approach, an OCR also works on sentence detection to preserve a document's structure.

The Text-to-Speech and Speech-to-Text are also used so that the details we speak are turned to digital data and the digitalized data is read to the user.

Keywords:OCR,Text-to-Speech,Speech-to-Text

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

OCR stands for ”Optical Character Recognition.” It is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents and images. OCR software can be used to convert a physical paper document, or an image into an accessible electronic version with text. For example, if you scan a paper document or photograph with a printer, the printer will most likely create a file with a digital image in it. The file could be a JPG/TIFF or PDF, but the new electronic file may still be only an image of the original document. You can then load this scanned electronic document it created, which contains the image, into an OCR program. The OCR program which will recognize the text and convert the document to an editable text file.

OCR software processes a digital image by locating and recognizing characters, such as letters, numbers, and symbols. Some OCR software will simply export the text, while other programs can convert the characters to editable text directly in the image. Advanced OCR software can export the size and formatting of the text as well as the layout of the text found on a page.

1.1.1 Motivation

In the old ways documentation was done by entering data manually using a keyboard. So to make it easier and faster for users to make documents.OCR was being used by people few years ago but not many people used it because it was expansive.So to make it less expansive and make it easier for people to get OCR's features this project was created.

1.2 Objective

The aim of this project is make text data to digital data. Thus the data can be documented. So that the retrieved data can be used in several places This application uses OCR in a way so that documentation can be done in a simple way. When we want to document something we type the content into the document using a keyboard which will take a lot of time. So to reduce the time needed and to make the documentation easy we use the OCR. The text-to-speech and speech-to-text are also used so that the details we speak are turned to digital data and the digitalized data is read to the user. We can select the documents which are read to the user using text-to-speech. The file is saved in docx format in the mobile device

1.3 Report Organization

The project report is divided into five sections.
Section 2 describes literature survey.
Section 3 describes the methodology used for implementing the project.
Section 4 gives the results and discussions.
Finally Section 5 gives the conclusion.

Chapter 2

Literature Survey

Early optical character recognition may be traced to technologies involving telegraphy and creating reading devices for the blind. In 1914, Emanuel Goldberg developed a machine that read characters and converted them into standard telegraph code. Concurrently, Edmund Fournier d'Albe developed the Optophone, a handheld scanner that when moved across a printed page, produced tones that corresponded to specific letters or characters.

In the late 1920s and into the 1930s Emanuel Goldberg developed what he called a ”Statistical Machine” for searching microfilm archives using an optical code recognition system. In 1931 he was granted USA Patent number 1,838,389 for the invention. The patent was acquired by IBM.

Character recognition is not a new problem but its roots can be traced back to systems before the inventions of computers. The earliest OCR systems were not computers but mechanical devices that were able to recognize characters, but very slow speed and lowaccuracy. In 1951, M. Sheppard invented a reading and robot GISMO that can be considered as the earliest work on modern OCR. GISMO can read musical notations as well as words on a printed page one by one. However, it can only recognize 23 characters. The machine also has the capability to could copy a typewritten page. J. Rainbow, in 1954, devised a machine that can read uppercase typewritten English characters, one per minute. The early OCR systems were criticized due to errors and slowrecognition speed. Hence, not much research efforts were put on the topic during 60's and 70's. The only developments were done on government agencies and large corporations like banks, newspapers and airlines etc. Because of the complexities associated with recognition, it was felt that three should be standardized OCR fonts for easing the task of recognition for OCR. Hence, OCRA and OCRB were developed by ANSI and EMCA in 1970, that provided comparatively acceptable recognition rates. During the past thirty years, substantial research has been done on OCR. This has lead to the emergence of document image analysis (DIA), multi-lingual, handwritten and omni-font OCRs Despite these extensive research efforts, the machine's ability to reliably read text is still far below the human. Hence, current OCR research is being done on improving accuracy and speed of OCR for diverse style documents printed/ written in unconstrained environments. There has not been availability of any open source or commercial software available for complex languages like Urdu or Sindhi etc.

Chapter 3

Methodology

3.1 Introduction

OCR stands for ”Optical Character Recognition.” It is a technology that recognizes text within a digital image. In this project the OCR uses Amazon Textract. Amazon Textract is a machine learning (ML) service that automatically extracts text, handwriting, and data from scanned documents. It goes beyond simple optical character recognition (OCR) to identify, understand, and extract data from forms and tables. Textract can extract the data in minutes instead of hours or days. Amazon Textract is a AWS(Amazon Web Services) AL services. AWS pre-trained AI services provide ready-made intelligence for your applications and workflows. Because we use the same deep learning technology that powers Amazon.com, you get quality and accuracy from continuously learning APIs. And best of all, AI services on AWS don’t require ML experience. Amazon Textract makes it easy to add document text detection and analysis to your applications. Using Amazon Textract customers can:

Detect typed and handwritten text in a variety of documents, including financial reports, medical records, and tax forms.

Extract text, forms, and tables from documents with structured data, using the Amazon Textract Document Analysis API.

Process invoices and receipts with the AnalyzeExpense API.

Process ID documents such as drivers licenses and passports issued by U.S. government, using the AnalyzeID API.

A Speech-to-Text API synchronous recognition request is the simplest method for performing recognition on speech audio data. Speech-to-Text can process up to 1 minute of speech audio data sent in a synchronous request. After Speech-to-Text processes and recognizes all of the audio, it returns a response.

Text-to-Speech create natural-sounding, synthetic human speech as playable audio. You can use the audio data files you create using Text-to-Speech to power your applications or augment media like videos or audio recordings (in compliance with the Google Cloud Platform Terms of Service including compliance with all applicable law).Text-to-Speech converts text or Speech Synthesis Markup Language (SSML) input into audio data like MP3.

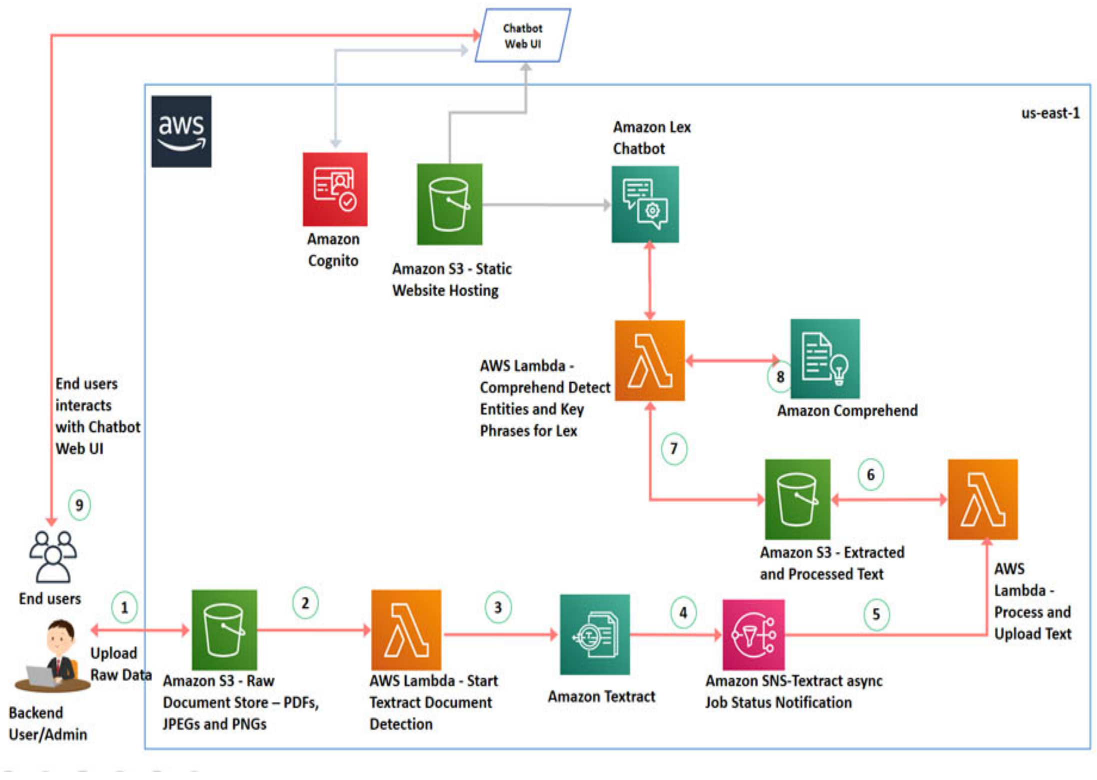

Chatbot Web UI
aws
us-east-1
Amazon Cognito
Amazon S3 - Static Website Hosting
Amazon Lex Chatbot
AWS Lambda - Comprehend Detect Entities and Key Phrases for Lex
8
Amazon Comprehend
7
End users interacts with Chatbot Web UI
9
6
Amazon S3 - Extracted and Processed Text
End users
1
2
3
4
5
AWS Lambda - Process and Upload Text
Upload Raw Data
Backend User/Admin
Amazon S3 - Raw Document Store – PDFs, JPEGs and PNGs
AWS Lambda - Start Textract Document Detection
Amazon Textract
Amazon SNS-Textract async Job Status Notification

Figure 3.1: Diagram illustrates the architecture of the solution

3.2 Modules

There is only one module in this application.

1.USER

1.Registration : The user can register in the application.

2.Login : The user can login into the application using username and password.

3.Scan files : The user uploads the image containg text to digital text.

4.Text to Speech : The text that is entered is converted to speech.

5.Speech to Text : The speech that is given is converted to text.

6.Save files : The files are saved in ’docx’ format in the phone.

7.View files : The ’docx’ files can be viewed using this application.

8.View user history : The user can view which document they have edited previously.

3.3 Developing Environment

1.Hardware Requirements

Processor - Intel x86

Speed - 1.1 GHz

RAM - 4 GB (min)

Hard Disk - 50 GB

2.Software Requirements

Operating System - Windows 7 or Above ,Android

Frontend - Android

Backend – Python ,MySQL

Platform used - PyCharm, Android Studio ,SQLyog

Web Browser - Google Chrome, Fire fox, Microsoft Edge

Frame work - Flask

3.4 Agile Methodology

The Agile methodology is a way to manage a project by breaking it up into several phases. It involves constant collaboration with stakeholders and continuous improvement at every stage. Once the work begins,teams cycle through a process of planning,executing and evaluating. Continuous collaboration is vital.

There is only one module in this project. The module User has the userstories:

- Has Low Priority: Login,Registration,View files,View history

- Has Medium Priority: Text to Speech

- Has High Priority: Scan files,Speech to Text,Save files

This project is divided into 4 sprints:

Sprint 1

-Table design

-Form design

-Coding

Sprint 2

-Scan files

-Text–to-speech

Sprint 3

-Speech-to-text

-Save files

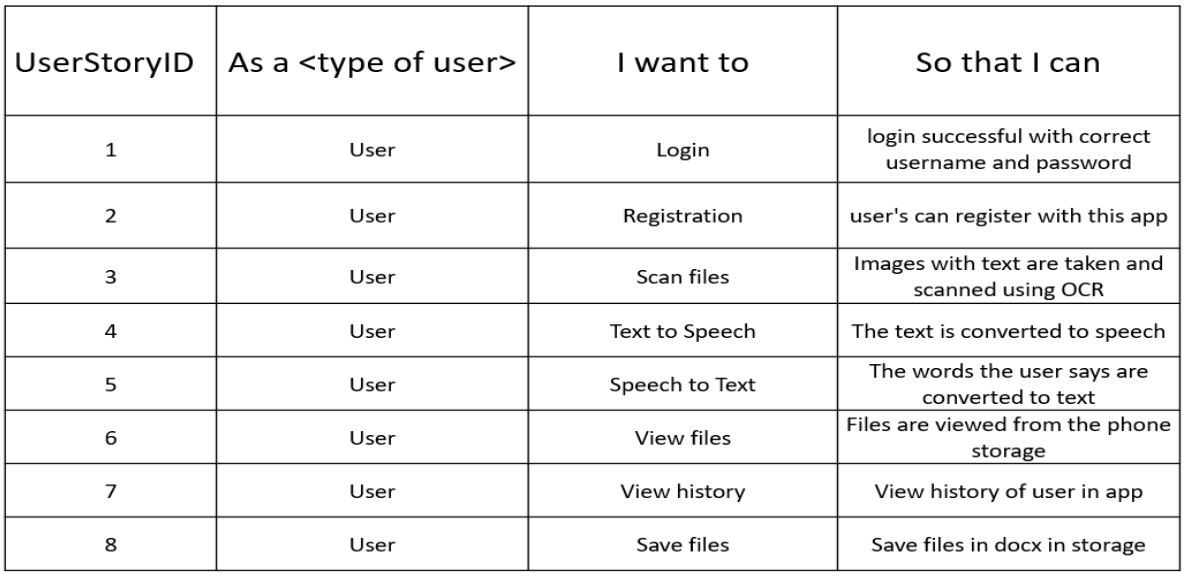
Sprint 4

-Testing data

-Output generation

3.4.1 User Story

A user story is a tool used in agile software development to capture a description of a software feature from an end-user perspective. The user story describes the type of user, what they want and why. A user story helps to create a simplified description of a requirement. User stories are one of the core components of an agile program. They help provide a user-focused framework for daily work drives collaboration, creativity, and a better product overall.There are 8 user stories.The user story is given below :

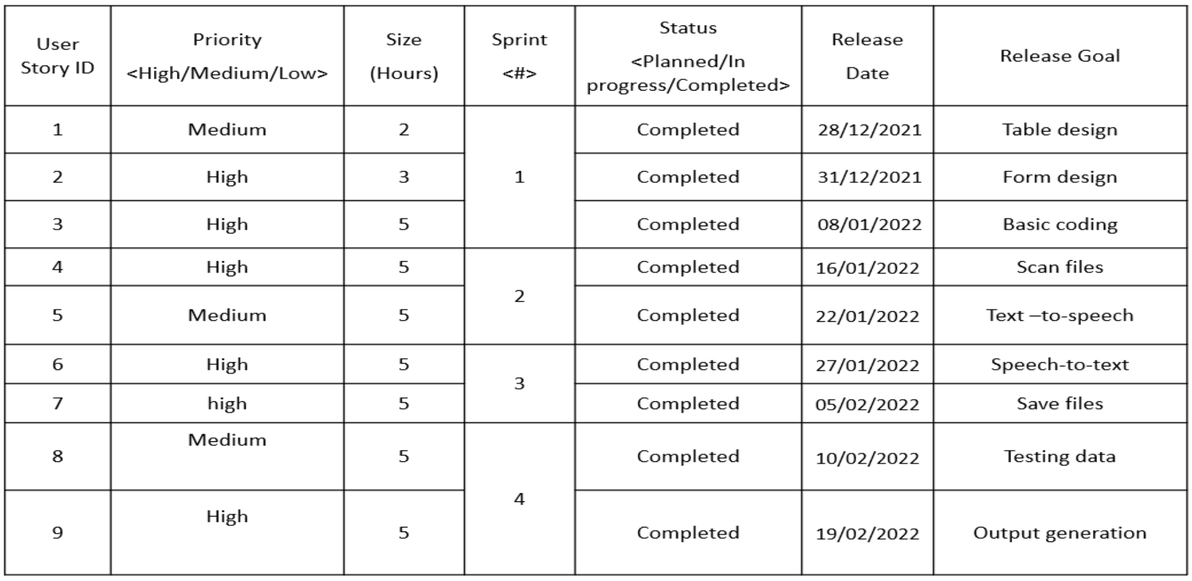


UserStoryID	As a <type of user>	I want to	So that I can
1	User	Login	login successful with correct username and password
2	User	Registration	user's can register with this app
3	User	Scan files	Images with text are taken and scanned using OCR
4	User	Text to Speech	The text is converted to speech
5	User	Speech to Text	The words the user says are converted to text
6	User	View files	Files are viewed from the phone storage
7	User	View history	View history of user in app
8	User	Save files	Save files in docx in storage

Figure 3.2: User Story

3.4.2 Product Backlog

A product backlog is a list of the new features, changes to existing features, bug fixes, infrastructure changes or other activities that a team may deliver in order to achieve a specific outcome.The priority of each user stories are given in the product backlog.

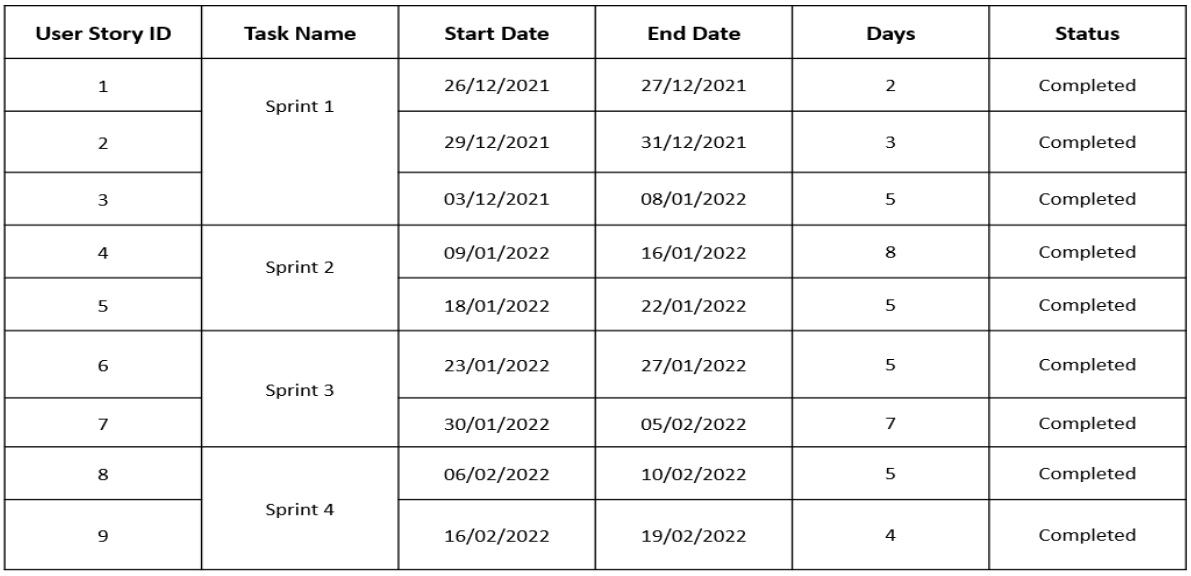


User Story ID	Priority <High/Medium/Low>	Size (Hours)	Sprint <#>	Status <Planned/In progress/Completed>	Release Date	Release Goal
1	Medium	2		Completed	28/12/2021	Table design
2	High	3	1	Completed	31/12/2021	Form design
3	High	5		Completed	08/01/2022	Basic coding
4	High	5	2	Completed	16/01/2022	Scan files
5	Medium	5		Completed	22/01/2022	Text –to-speech
6	High	5	3	Completed	27/01/2022	Speech-to-text
7	high	5		Completed	05/02/2022	Save files
8	Medium	5	4	Completed	10/02/2022	Testing data
9	High	5		Completed	19/02/2022	Output generation

Figure 3.3: Product Backlog

3.4.3 Project Plan

A project plan that has a series of tasks laid out for the entire project, listing task durations, responsibility assignments, and dependencies. Plans are developed in this manner based on the assumption that the Project Manager, hopefully along with the team, can predict up front everything that will need to happen in the project, how long it will take, and who will be able to do it.The project plan shows when each sprint starte

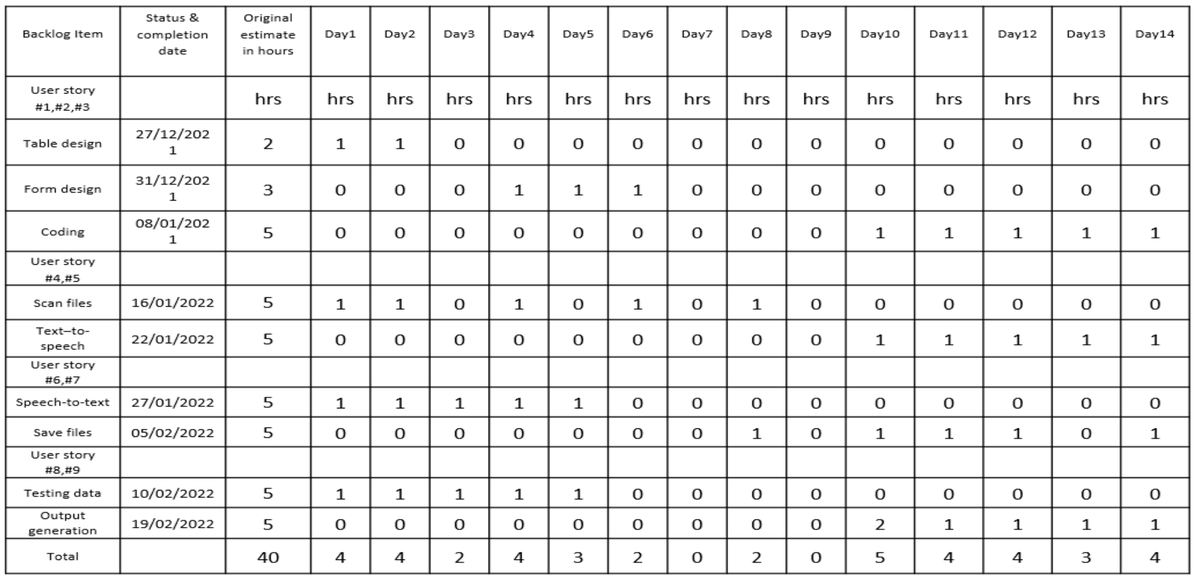


User Story ID	Task Name	Start Date	End Date	Days	Status
1	Sprint 1	26/12/2021	27/12/2021	2	Completed
2		29/12/2021	31/12/2021	3	Completed
3		03/12/2021	08/01/2022	5	Completed
4	Sprint 2	09/01/2022	16/01/2022	8	Completed
5		18/01/2022	22/01/2022	5	Completed
6	Sprint 3	23/01/2022	27/01/2022	5	Completed
7		30/01/2022	05/02/2022	7	Completed
8	Sprint 4	06/02/2022	10/02/2022	5	Completed
9		16/02/2022	19/02/2022	4	Completed

Figure 3.4: Project Plan

3.4.4 Sprint Plan

The sprint backlog is a list of tasks identified by the Scrum team to be completed during the Scrum sprint. During the sprint planning meeting, the team selects some number of product backlog items, usually in the form of user stories, and identifies the tasks necessary to complete each user story.Here the sprint backlog this project is given below :

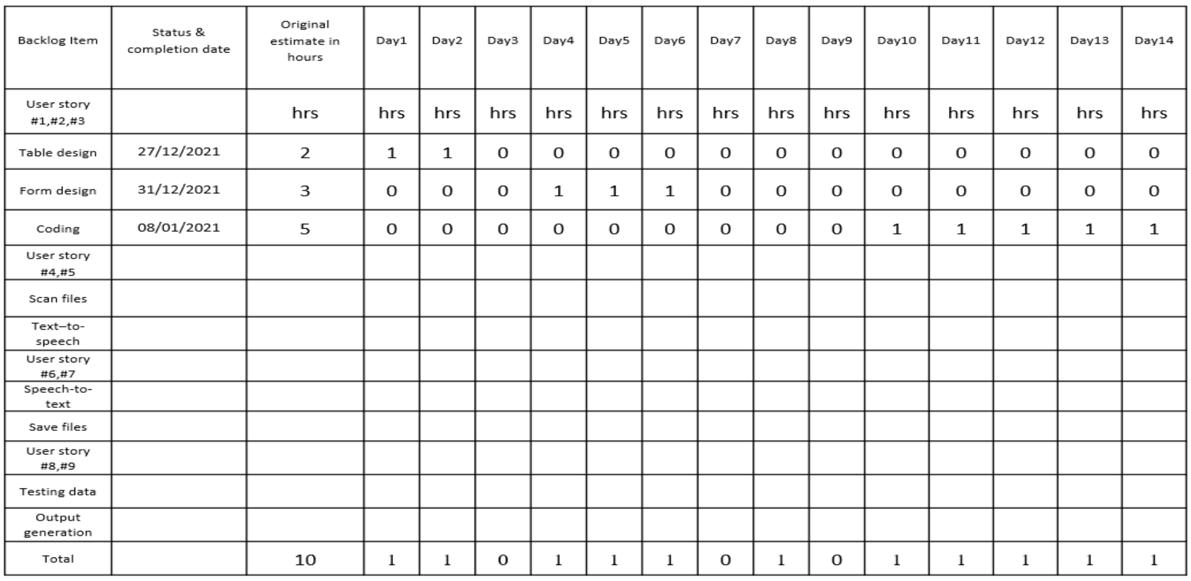


Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	27/12/202 1	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/202 1	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Coding	08/01/202 1	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #4,#5																
Scan files	16/01/2022	5	1	1	0	1	0	1	0	1	0	0	0	0	0	0
Text–to-speech	22/01/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #6,#7																
Speech-to-text	27/01/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Save files	05/02/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
User story #8,#9																
Testing data	10/02/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Output generation	19/02/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Total		40	4	4	2	4	3	2	0	2	0	5	4	4	3	4

Figure 3.5: Sprint Plan

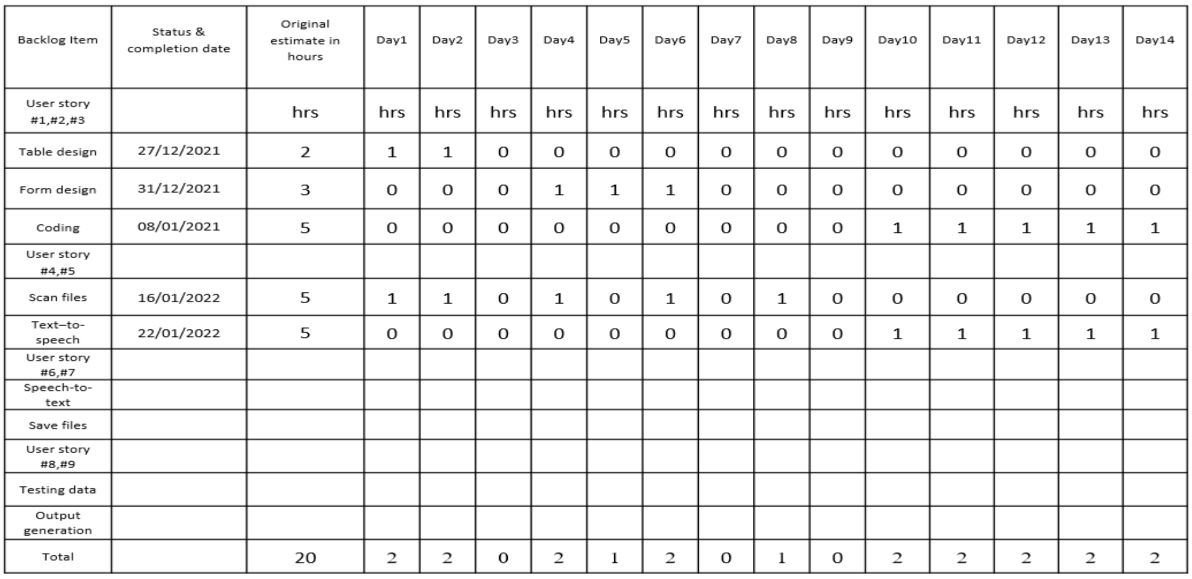
3.4.5 Sprint Actuals

The sprint actual is done based on the sprint plan. The sprint actual is divided into 4 sprints in which each of the tasks specified in the sprint backlog are done based on each sprint.Each sprint needs to be completed based on the dates in the sprint backlog.



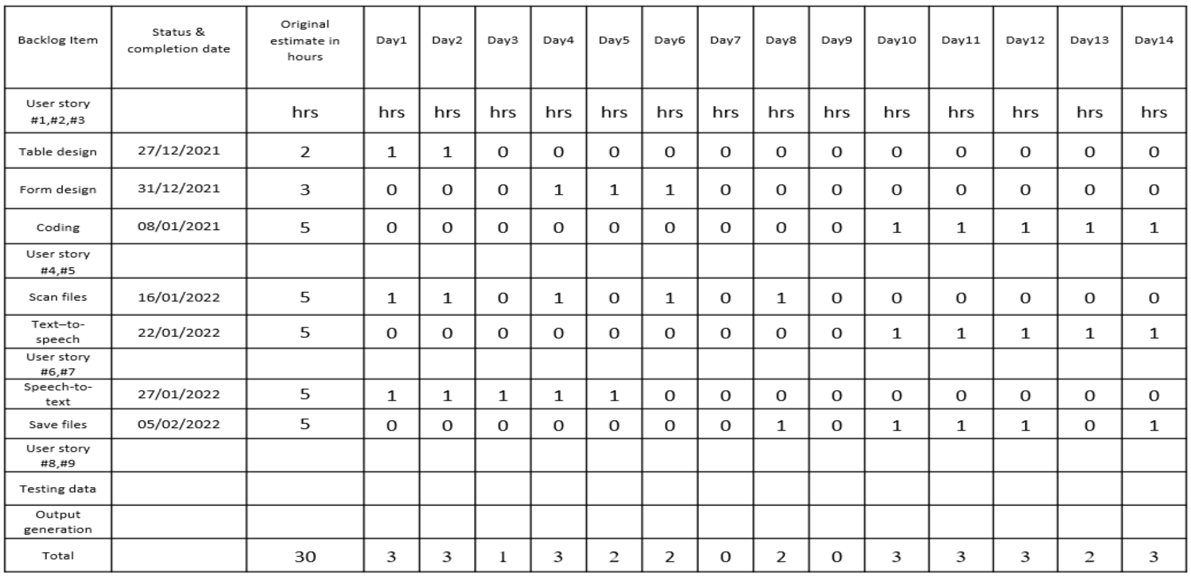
Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	27/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Coding	08/01/2021	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #4,#5																
Scan files																
Text–to-speech																
User story #6,#7																
Speech-to-text																
Save files																
User story #8,#9																
Testing data																
Output generation																
Total		10	1	1	0	1	1	1	0	1	0	1	1	1	1	1

Figure 3.6: Sprint 1 Actual



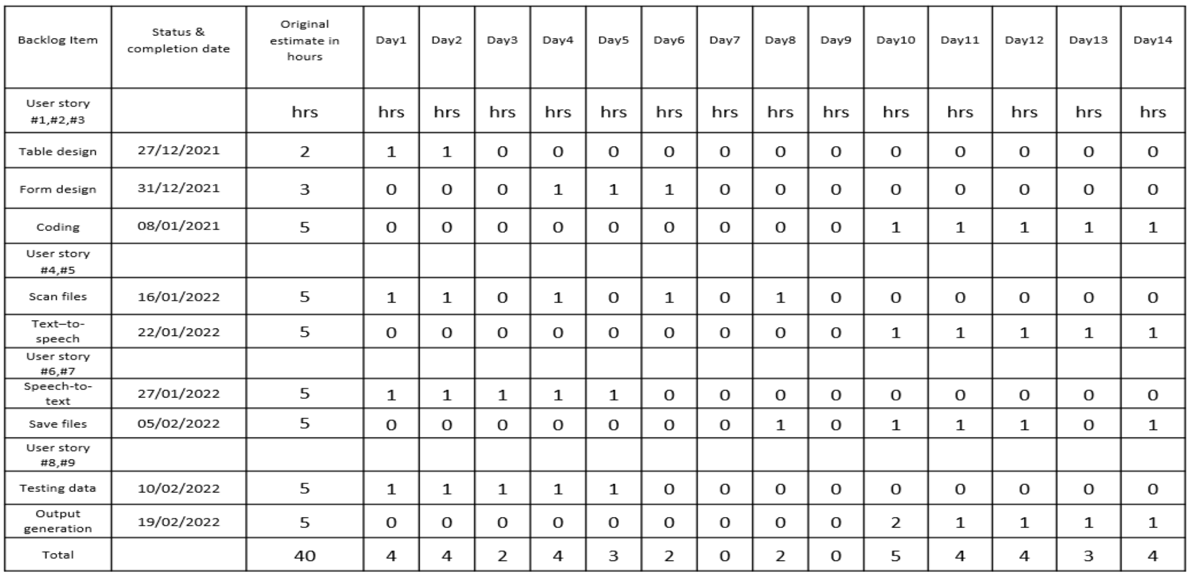
Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	27/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Coding	08/01/2021	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #4,#5																
Scan files	16/01/2022	5	1	1	0	1	0	1	0	1	0	0	0	0	0	0
Text–to-speech	22/01/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #6,#7																
Speech-to-text																
Save files																
User story #8,#9																
Testing data																
Output generation																
Total		20	2	2	0	2	1	2	0	1	0	2	2	2	2	2

Figure 3.7: Sprint 2 Actual



Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	27/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Coding	08/01/2021	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #4,#5																
Scan files	16/01/2022	5	1	1	0	1	0	1	0	1	0	0	0	0	0	0
Text–to-speech	22/01/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #6,#7																
Speech-to-text	27/01/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Save files	05/02/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
User story #8,#9																
Testing data																
Output generation																
Total		30	3	3	1	3	2	2	0	2	0	3	3	3	2	3

Figure 3.8: Sprint 3 Actual



Backlog Item	Status & completion date	Original estimate in hours	Day1	Day2	Day3	Day4	Day5	Day6	Day7	Day8	Day9	Day10	Day11	Day12	Day13	Day14
User story #1,#2,#3		hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs	hrs
Table design	27/12/2021	2	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Form design	31/12/2021	3	0	0	0	1	1	1	0	0	0	0	0	0	0	0
Coding	08/01/2021	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #4,#5																
Scan files	16/01/2022	5	1	1	0	1	0	1	0	1	0	0	0	0	0	0
Text–to-speech	22/01/2022	5	0	0	0	0	0	0	0	0	0	1	1	1	1	1
User story #6,#7																
Speech-to-text	27/01/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Save files	05/02/2022	5	0	0	0	0	0	0	0	1	0	1	1	1	0	1
User story #8,#9																
Testing data	10/02/2022	5	1	1	1	1	1	0	0	0	0	0	0	0	0	0
Output generation	19/02/2022	5	0	0	0	0	0	0	0	0	0	2	1	1	1	1
Total		40	4	4	2	4	3	2	0	2	0	5	4	4	3	4

Figure 3.9: Sprint 4 Actual

Chapter 4

Results and Discussions

4.1 Results

The text in the image is extracted by the OCR and the text is converted to digital text.The OCR can be improved by giving it quality images so that the OCR can find the text easily.The Text-to-Speech Speech-to-Text gives the outputs and it can be stored in the device.The data we retrieve using this app is be stored in 'docx' format.

1.The user can first login in the app if the user is registered before.

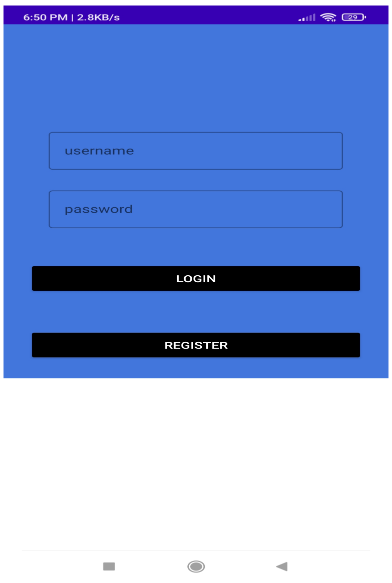


Figure 4.1: Login

2.If the user is not registered than the user can register by clicking on the register button in the login page. then the registration page is shown.

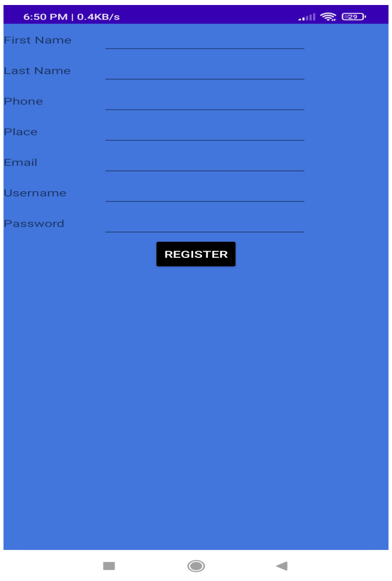


Figure 4.2: Register

3.After the user register's or login's the user is taken to the homepage of the application.There the user can select the features the user needs.

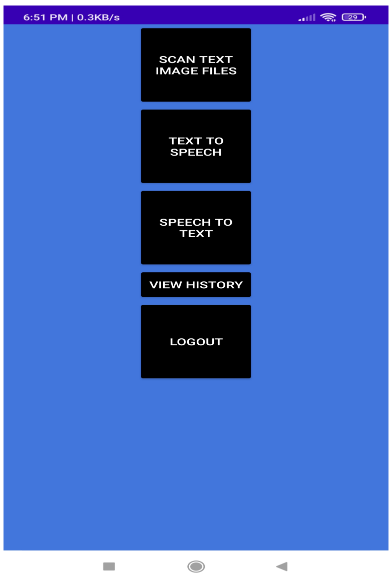


Figure 4.3: Homepage

4.The user selects the feature ’scan text image files’ ,then the user is taken to the image upload page where the user clicks on the browse button and goes to the gallery of the phone to select the image the user needs.

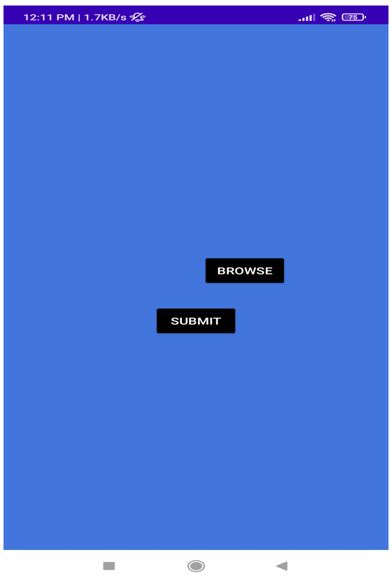


Figure 4.4: Image Upload-Before Uploading

5.After the image is selected the image will appear in the image upload page and the user clicks the submit button.

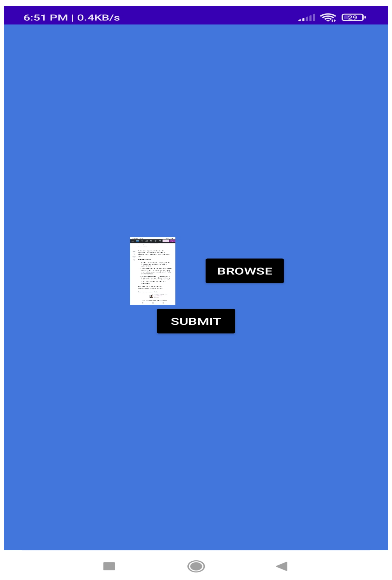
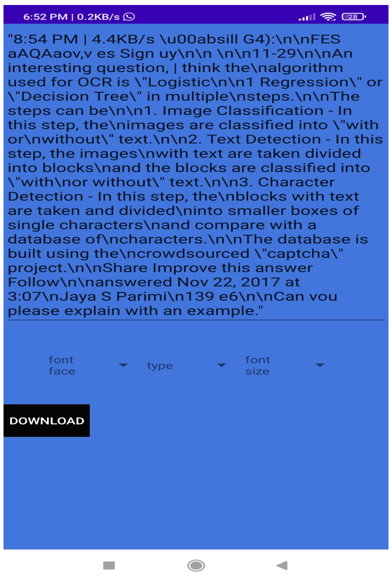


Figure 4.5: Image Upload-After Uploading

6.Then the extracted text is shown in the result page.Then the user can edit the text if they want to.Then the user can save the text in ’docx’ format.



```
6:52 PM | 0.2KB/s
"8:54 PM | 4.4KB/s \u00absill G4):\n\nFES
aAQAaov,v es Sign uy\n\n \n\n11-29\n\nAn
interesting question, | think the\nalgorithm
used for OCR is \"Logistic\n\n1 Regression\" or
\"Decision Tree\" in multiple\nsteps.\n\nThe
steps can be\n\n1. Image Classification - In
this step, the\nimages are classified into \"with
or\nwithout\" text.\n\n2. Text Detection - In this
step, the images\nwith text are taken divided
into blocks\nand the blocks are classified into
\"with\nor without\" text.\n\n3. Character
Detection - In this step, the\nblocks with text
are taken and divided\ninto smaller boxes of
single characters\nand compare with a
database of\ncharacters.\n\nThe database is
built using the\ncrowdsourced \"captcha\"
project.\n\nShare Improve this answer
Follow\n\nanswered Nov 22, 2017 at
3:07\nJaya S Parimi\n139 e6\n\nCan vou
please explain with an example."
```

Figure 4.6: Result

7.The user can select the ’text to speech’ feature,then it will take the user to the page with upload a file and direct text to speech function.

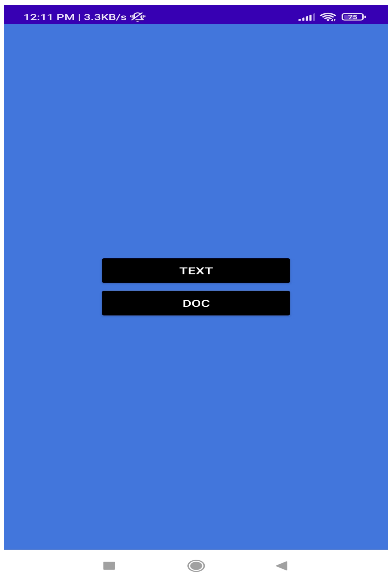


Figure 4.7: Text to Speech-1

8.This page is where user enters the text to be converted to speech.

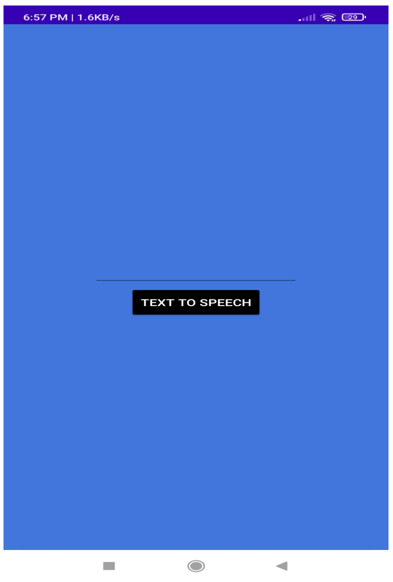


Figure 4.8: Text to Speech-2

9.This page is where the audio is converted to text and it can be saved.The user can speak after clicking the record button and the stop by clicking the stop button.Then it is saved in ’docx’ format.

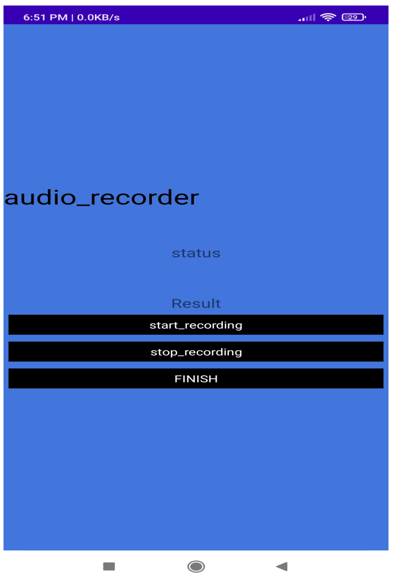


Figure 4.9: speech to Text

4.2 Discussions

OCR is finally moving away from just seeing and matching. Driven by deep learning, it's entering a new phase where it first recognizes scanned text, then makes meaning of it. The competitive edge will be given to the software that provides the most powerful information extraction and highest-quality insights. Now OCR can only recognition characters, it can be modified to identify shapes ,flowcharts ,diagrams ,graphs ,etc in the future.

OCR technology is becoming very widespread in professional environments such as historical archives, museums, and libraries. This is a great way to preserve ancient texts or images in a digital format. More importantly, these documents can also be examined in the digital domain without disturbing the original physical materials.

Chapter 5

Conclusions

In conclusion, OCR is a very remarkable technology that holds a lot of potential. In this day and age, such tools are already quite advanced. However, Optical Character Recognition is going to look even better in the future. AI is on the way to becoming one of the most influential trends in the coming years, revolutionizing information as we know it.

OCR will continue to be a valuable tool for filling in gaps whereby an application-generated electronic document cannot be generated. Ultimately, the truly “paperless business” doesn’t (yet) exist and data extraction is still a useful tool that can augment e-document processing.

References

[1] **Dhavale, Sunita Vikrant** (2017) . Advanced Image-Based Spam Detection and Filtering Techniques, Hershey, PA: IGI Global. p. 91. ISBN 9781683180142. Retrieved September 27, 2019.

[2] **Stephen V. Rice et al. Kluwer Academic** (1999) Optical Character Recognition: An Illustrated Guide to the Frontier

[3] **Mohamed Cheriet, Nawwaf Kharma, Cheng-Lin Liu. John Wiley Sons** (2007) Character Recognition Systems: A Guide for Students and Practitioners.

[4] **Herbert F Schantz, Recognition Technologies Users Association** (1982) The History of OCR: Optical Character Recognition.

Appendix

Source Code

```
   # Copyright (c) 2022 KT VIMAL
   # Code written by : KT VIMAL
   # Email ID : vimalvipanjika@gmail.com

   import os

import datetime

import cv2
import pytesseract
from PIL import Image
from flask import *

from werkzeug.utils import secure_filename
import speech_recognition as sr

from src.dbconnect import *
app=Flask(__name__)
import docx
from docx.shared import Pt


def docc1(text,ftype,fface,fsize):
   doc = docx.Document()

   text=text.split('\\n')
   print(len(text))
   for i in text:
      doc_para = doc.add_paragraph('')

      if ftype=='bold':
         doc_para.add_run(i).bold = True
      elif ftype=='italic':
         doc_para.add_run(i).italic = True
      elif ftype == 'all':
         d=doc_para.add_run(i)
         d.bold = True
         d.italic = True
         print("okkkkkkkkkkkkkk")

      else:
         doc_para.add_run(i)

   Tstyle = doc.styles['Normal']
```

Appendix

```
   font = Tstyle.font
   font.name = fface
   font.size = Pt(fsize)
   import time

   fn = time.strftime("%Y%m%d_%H%M%S") + "." + ".doc"


   doc.save('static/texttodoc/'+fn)
   return fn
@app.route('/imgtotext2',methods=['post'])
def imgtotext2():
   print(request.form)
   text=request.form['text']
   ftype=request.form['ftype']
   fface=request.form['fface']
   fsize=request.form['fsize']
   id=request.form['id']


   res=docc1(text,ftype,fface,int(fsize))
   print("rrrrrr",res)
   qry="insert into history values(null,%s,%s,curdate())"
   val=(id,res)
   iud(qry,val)
   return jsonify({"task":res})
@app.route('/imgtotext1',methods=['post'])
def imgtotext1():
   text=request.form['text']
   ftype=request.form['ftype']
   fface=request.form['fface']
   fsize=request.form['fsize']
   image = request.files['files']
   image.save("static/texttodoc/sampe.png")
   print(image)
   img=cv2.imread("static/texttodoc/sampe.png")
   imgg= cv2.resize(img, (400, 400),
           interpolation = cv2.INTER_NEAREST)
   cv2.imwrite("static/texttodoc/sampe1.png",imgg)
   res=docc(text,ftype,fface,int(fsize))

   return jsonify({"task":res})
def docc(text,ftype,fface,fsize):
   doc = docx.Document()

   text=text.split('\\n')
   print(len(text))
   for i in text:
      doc_para = doc.add_paragraph('')

      if ftype=='bold':
         doc_para.add_run(i).bold = True
      elif ftype=='italic':
         doc_para.add_run(i).italic = True
      elif ftype == 'all':
         d=doc_para.add_run(i)
         d.bold = True
         d.italic = True
         print("okkkkkkkkkkkkk")

      else:
         doc_para.add_run(i)
   doc.add_picture("static/texttodoc/sampe1.png")
```

Appendix

```
   Tstyle = doc.styles['Normal']
   font = Tstyle.font
   font.name = fface
   font.size = Pt(fsize)
   fn = datetime.now().strftime("%Y%m%d%H%M%S") + ".doc"

   doc.save('static/texttodoc/'+fn)
   return fn
@app.route("/logincode",methods=['post'])
def logincode():
   uname=request.form['username']
   password=request.form['password']
   q="select * from login where username=%s and password=%s "
   val=uname,password
   res=selectonecond(q,val)
   if res is None:
      return jsonify({'task': 'invalid'})
   else:
      return jsonify({'task': 'success','lid':res[0],'type':res[3]})




@app.route("/register",methods=['post'])
def register():
   fname=request.form['fname']
   lname=request.form['lname']
   Phone=request.form['Phone']
   Place=request.form['Place']
   Email=request.form['Email']
   username=request.form['username']
   password=request.form['password']
   qry = "insert into login values(null,%s,%s,'user')"
   value = (username, password)
   lid = iud(qry, value)
   qry1 = "insert into user values(null,%s,%s,%s,%s,%s,%s)"
   val = (str(lid), fname, lname,Place,Phone,Email)
   iud(qry1, val)

   return jsonify({'task': 'success'})

@app.route('/doctospeech',methods=['post'])
def doctospeech():


   file = request.files['file']
   ff = secure_filename(file.filename)


   import time

   req = time.strftime("%Y%m%d_%H%M%S") + "." +".doc"
   print(req)
   file.save(os.path.join('./static/doctospeech', req))

   import docx2txt
   result = docx2txt.process("./static/doctospeech/"+req)
   print(result)
   return jsonify({'task':result})


@app.route('/imgtotext',methods=['post'])
def imgtotext():
```

Appendix

```
    print(request.form)
    image = request.files['files']
    print(image)
    img = secure_filename(image.filename)

    image.save(os.path.join("./static/imgtotext", img))
    text=main(os.path.join("./static/imgtotext", img))
    print(text,"=============================================")
    return jsonify(text)
def main(path):

    # Load the required image
    print("path===",path)
    image = cv2.imread(path)
    # print(image)
    gray = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)

    # Store grayscale image as a temporary file to apply OCR
    filename = "{}.png".format("temp")
    cv2.imwrite(filename, gray)

    # Load the image as a PIL/Pillow image, apply OCR, and then delete the temporary file
    pytesseract.pytesseract.tesseract_cmd = r'C:\Program Files\Tesseract-OCR\tesseract.exe'
    text = pytesseract.image_to_string(Image.open(filename))

    print("OCR Text is " + text.strip())
    return text.strip()


@app.route('/photoupload', methods=['POST','GET'])
def photoupload():
    try:
        # print(request.form)
        print(request.files)


        file=request.files['file']
        ff=secure_filename(file.filename)
        print(ff)
        fl = ff.split('.')
        print(fl[1])
        import time
        ffl=time.strftime("%Y%m%d_%H%M%S")
        req = time.strftime("%Y%m%d_%H%M%S") + "." + fl[1]
        file.save(os.path.join('./static/audio', req))


        os.system('ffmpeg -i E:\\final\\OCR\\src\\static\\audio\\'+req+' E:\\final\\OCR\\src\\static\\audio\\'+ffl+".wav")
        #
        res=silence_based_conversion(ffl)
        print(res)


        return jsonify({'task': res})


    except Exception as e:
        print(e)

        return jsonify({'task':"No result"})


def silence_based_conversion(fl):
     path = r'static/audio/'+fl+'.wav'



     r = sr.Recognizer()
     file=path
     # recognize the chunk
     with sr.AudioFile(file) as source:
        # remove this if it is not working
        # correctly.
        r.adjust_for_ambient_noise(source)
        audio_listened = r.listen(source)

     try:
        # try converting it to text
        rec = r.recognize_google(audio_listened)
        # write the output to the file.
        print(rec)
        return rec


     except sr.UnknownValueError:
        print("Could not understand audio")

     except sr.RequestError as e:
        print("Could not request results. check your internet connection")

  @app.route('/texttodoc',methods=['post'])
  def texttodoc():
     text= request.form['text']
     print(text)
     import docx
     doc = docx.Document()
     doc_para = doc.add_paragraph(text)
     print(doc_para)

     import time

     req = time.strftime("%Y%m%d_%H%M%S") + "." +".doc"
     doc.save(os.path.join('./static/texttodoc/', req))
     print(req)


     # pip install python-docx
     return jsonify({'task': req})





  @app.route("/history",methods=['post'])
  def history():
     print(request.form)
     uid=request.form['uid']
     qry="SELECT * from `history` where uid=%s"
     val=uid
     res = androidselectall(qry,val)
     print(res)
     return jsonify(res)

  if __name__=='__main__':
     app.run(host='0.0.0.0',port=5000)
```

Database Design

Attribute Name	**Datatype**	**Width**	**Description**
id	Integer	11	Primary Key
lid	Integer	11	
fname	Varchar	45	
lname	Varchar	45	
place	Varchar	45	
phone	Varchar	45	
email	Varchar	54	

Table A.1: user

Attribute Name	**Datatype**	**Width**	**Description**
id	Integer	11	Primary Key
username	Varchar	45	
password	Varchar	45	
type	Varchar	45	

Table A.2: login

Attribute Name	**Datatype**	**Width**	**Description**
id	Integer	11	Primary Key
uid	Integer	11	
file	Varchar	54	
date	Varchar	54	

Table A.3: history

DataFlow Diagram

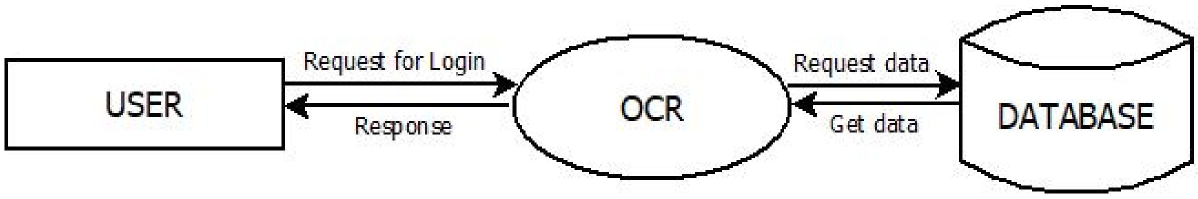


Figure A.1: Level 0

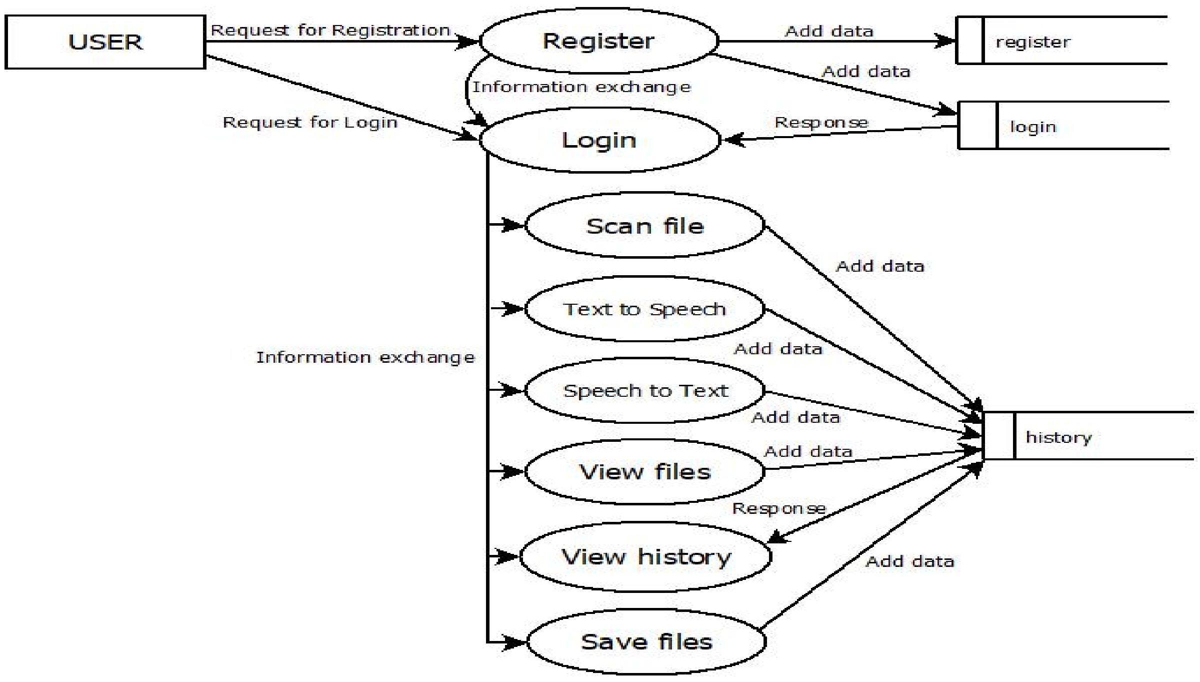


Figure A.2: Level 1